

User Manual



Version	1.2
Date	30/02/2013
Authors	S.Wood, R.Knight
Institution	Sheffield Teaching Hospitals

Contents

1	<u>OVERVIEW</u>	<u>3</u>
2	<u>CONNECT TO A DATA SOURCE</u>	<u>5</u>
2.1	TEXT BASED SOURCES.....	6
2.1.1	DELIMITED DATA FILE	6
2.1.2	EXCEL DATA FILES.....	7
2.1.3	CSV FILE COLLECTION	8
2.2	DATABASE SOURCES	9
2.2.1	MICROSOFT SQL, MYSQL AND ARQ	9
2.2.2	MICROSOFT ACCESS DATABASE	10
2.2.3	MICROSOFT OLE DB PLUGIN.....	11
2.3	SYSTEM SPECIFIC SOURCES.....	12
2.3.1	CLEARCANVAS WORKSTATION IMAGES	12
2.3.2	XNAT IMPORTER	13
3	<u>WORKING WITH TABLES AND FIELDS</u>	<u>14</u>
3.1	SOURCE PANEL COLOUR CODING.....	14
3.2	RENAMING	14
3.3	KEYS.....	15
3.4	RELATIONSHIPS.....	16
4	<u>ANNOTATION</u>	<u>18</u>
4.1	SEMANTIC DATA ANNOTATION.....	18
4.2	SEMANTIC DATA TRANSFORMATION	20
4.3	VIEW THE TABLE CONTENTS	21
4.4	VIEW DATA TYPES	22
5	<u>CREATING A NEW DESTINATION</u>	<u>23</u>
6	<u>DE-IDENTIFYING THE DATASET</u>	<u>25</u>
6.1	TABLE OPTIONS	25
6.2	FIELD OPTIONS	25
6.3	FILE OPTIONS	27
7	<u>DATASET PROPERTIES.....</u>	<u>28</u>
7.1	METADATA/SEARCH PROPERTIES.....	30
7.2	ACCESS CONTROL	30
8	<u>DATA PUBLICATION</u>	<u>31</u>
9	<u>QUERY THE WEB DATASET</u>	<u>32</u>

1 Overview

The Data Publication Suite is designed to support the process of publishing, in a secure internet accessible way, clinical or research data sets. When we use the term data sets we primarily refer to structured, and potentially relational, data sources such as CSV extracts and relational databases. Once published the data is made accessible through [SPARQL](#) and an SQL type protocol from the [OGSA-DIA](#) project.

The general process for publication, although many of them are not mandatory, is as follows:

- Import a data source
- Define relationships between the tables if they exist and are not automatically detected
- Semantically annotate the data
- Create a destination container of the server (if you have permissions)
- Create a new destination based on a data source
- Define a de-identification profile for this destination
- Publish the data
- Manage the access list for the resource

The DPS contains a large number of features for managing all of these requirements and the document will go through the each of these in turn.

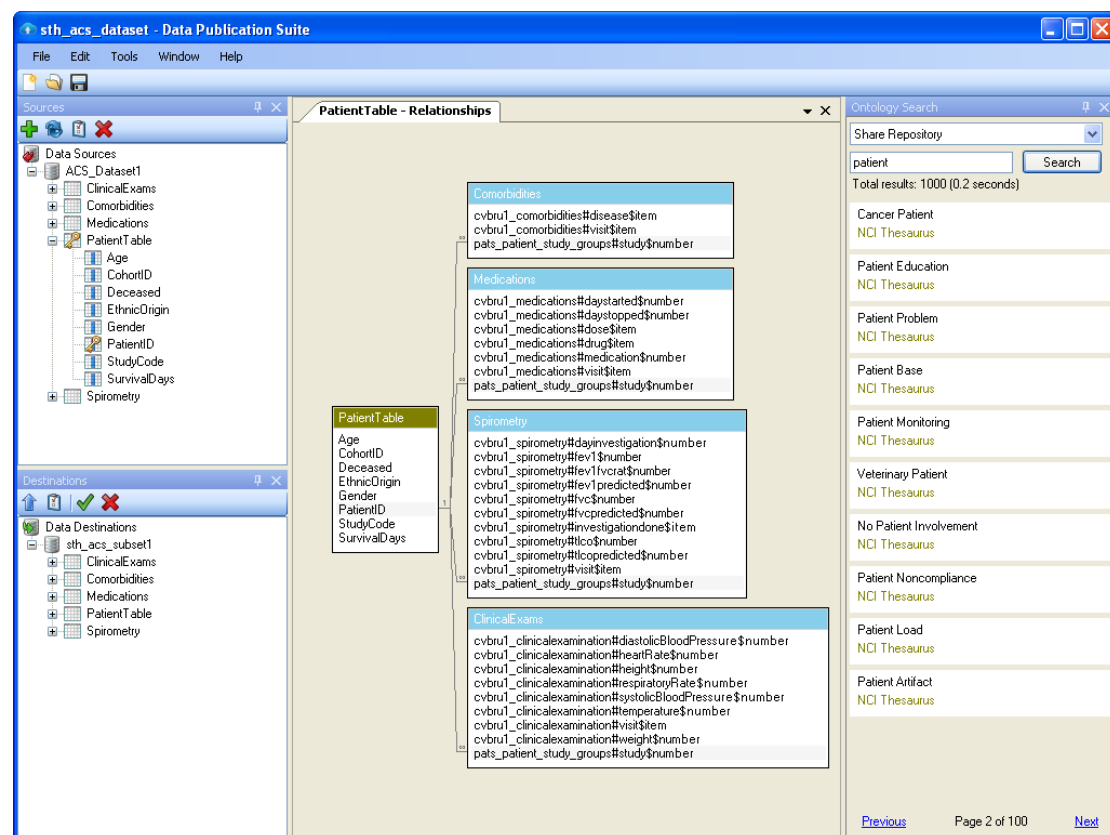


Figure 1 General application display

Figure 1 shows a general screen shot of the software running, with a central tabbed panel describing the table relationships.

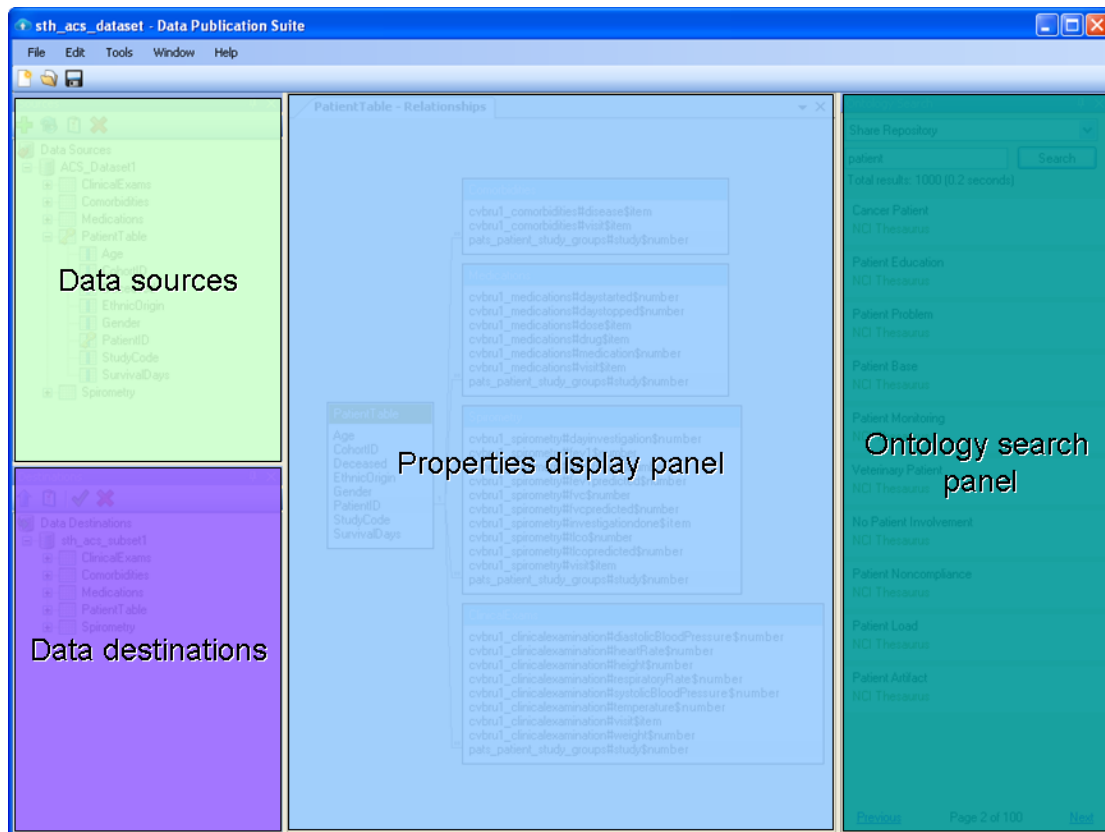


Figure 2 Application functional panels

Figure 2 shows the structural components of the interface, each of these will be explored in detail during the document but as an overview the basic functionality is as follows.

Data Sources: The system is designed to allow you manage multiple data sources within the same project, although this can lead to some confusion if you intend to have multiple destination per source so normally we would only manage single source within a project. This panel allows you to manage the data import, structure of the data, and also the semantic annotation of the data source.

Data Destinations: Each source can have multiple destinations and this are is where these are listed and managed. In this area you can create a profile for the destination which also provides data transformation operations like withhold data items, or process them in some way to de-identify the original source in some way.

Ontology search: A single Google type search window which allows the user to find semantic terms to annotate data. Two query destination appear by default, these are bioportal and the specialised VPH-Share repository. Both function but VPH-Share is far better so we would normally use this by default. Terms from this window can simply be dragged onto data items in the sources window to perform annotation.

Properties panel: This is a tabbed area where a other information is displayed. The example in Figure 1 shows the relationship manager but other tabs include the metadata management for each destination of the specific field properties for a data item.

As a design philosophy this software makes heavy use of context menus so if you think there should be an option to do something at any point in the application please try to right click.

2 Connect to a Data Source

As has been previously mentioned, the first step in publishing data is to connect to locally available data which the DPS refers to as a data source.

Once the DPS application has been started, in the data source window the user should click the add source button as indicated in Figure 3, which opens the new source window. The left section of this window lists the installed source types which can be selected and the right section displays the configuration of the selected source.

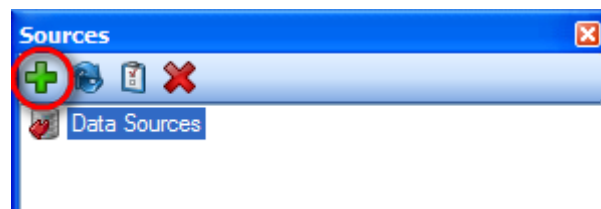


Figure 3. Highlighting the add source button.

2.1 Text Based Sources

2.1.1 Delimited data file

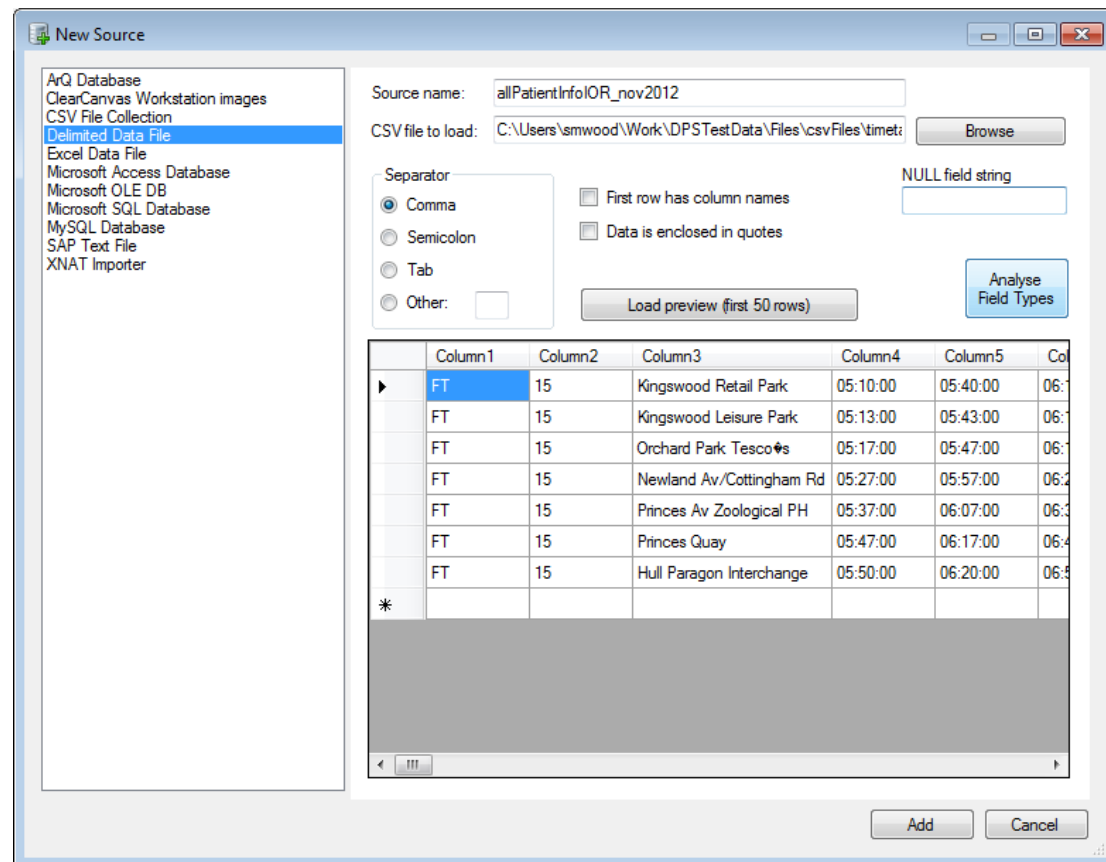


Figure 4 Delimited data source configuration

This plugin is one of the core plugins of the system, and is also one of the most difficult to develop since data can be provided in hugely varied forms. The interface is relatively self-evident in its use but there are steps that you should perform before proceeding with the data import.

First, always load the preview to see what the system makes of the file provided, it attempts to “guess” what the delimiters are but this can be incorrect depending of the file format.

Second check the data types the plugin has assigned to the data. There are many ways in which the data can be interpreted and in order for you to use the data effectively via the cloud services it needs to be in the correct format.

The “NULL field string” option tells the system to leave an entry in the database truly empty if it contains this string, in database parlance this is called a null entry. What can often happen when exporting from another system is that empty entries can have the string “null”, “NULL” or even “Empty” in them. It is not a problem to publish data with these entries but when someone tries to query the data they need to know that instead of asking for empty entries they need ask for entries equal to “null” or whatever the value is. It is worth noting that querying for empty entries in a database is very common scenario as it quite often indicates that something has not been done and action is required and so it is important that the publisher and consumers of the data have the same understanding on this concept.

2.1.1.1 Handling dates and times

In particular dates and times are very problematic as there are a large number of different formats that they can be provided in. For instance 01/06/2014 in the US would be translated to the 6th of Jan 2014 and in Europe would be 1st June 2014. There are system settings within windows to control the “culture” of the operating system but these often are not set appropriately so you should always check. If you have any control over the data that is contained in the file the most unambiguous form for a date is of the form “dd mmm yyyy” e.g. 17 Dec 1971 and for times it is “hh:mm:ss” using 24 hour clock e.g. 22:01:55. The reverse form for dates is also unambiguously interpreted “yyyy/mm/dd” e.g. 1971/12/17.

Dates and times are probably the most important type to check properly since they are the sources of most of the problems we have encountered in the data management process.

2.1.2 Excel Data Files

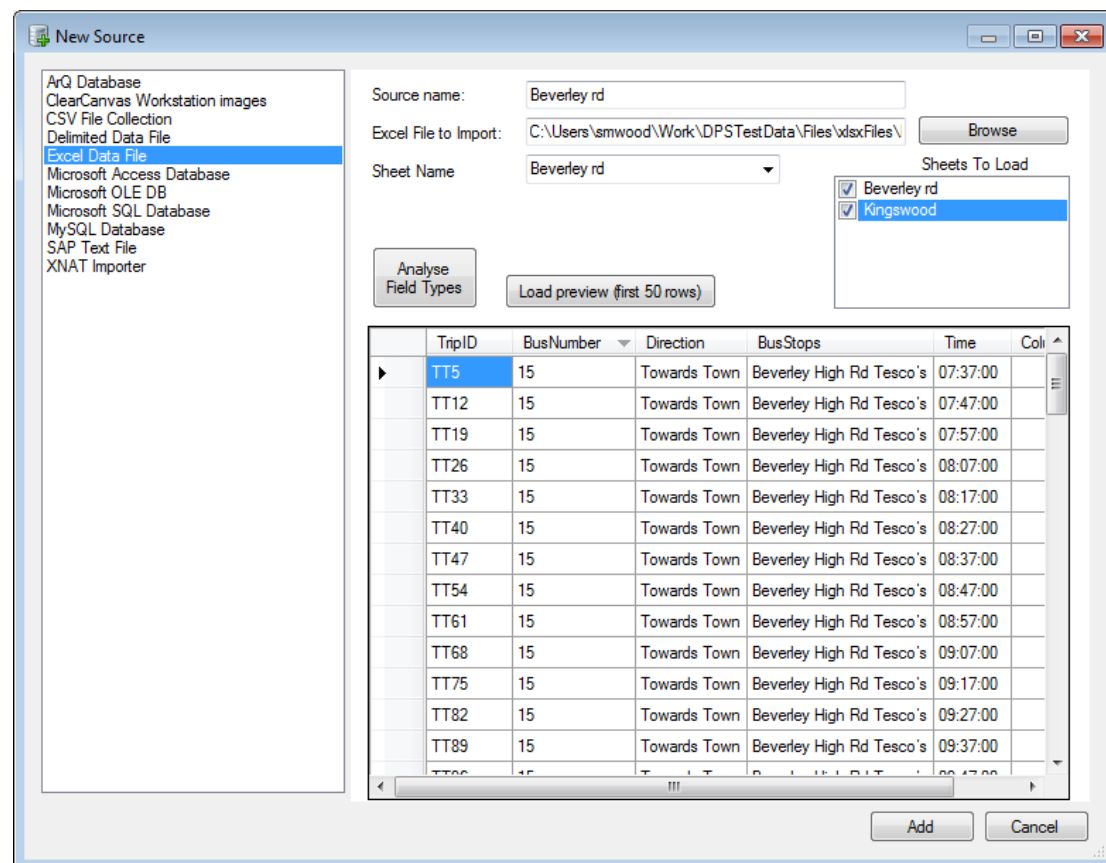


Figure 5 Excel data source configuration

This has a very similar form to the delimited file plugin in terms of configuration options you simply select the excel spreadsheet. The system should then, where it is possible, produce a table from each of the sheets within the files. These sheets will be the equivalent of tables in the DPS.

You can inspect the contents of the sheets by selecting one from the dropdown menu and clicking “Load preview”. There is also an option for you to select which of the sheets in the file you wish to import into the DPS.

Since the process behind the scenes it to ask Excel to export the required sheets to CSV file format and then these are loaded into the DPS the caveats and advice on checking the data typing once imported still hold.



NOTE: This plugin requires that Microsoft Office is installed on the machine in order to work. We have found significant differences in how versions of Office handle this type of remote control and whilst we have tested as many as we have access to it is possible the process will fail on some installations. In this case the fall back would be to manually use Excel to export the sheets to csv files and use either the “CSV File Collection” if there are multiple sheets to be imported or the “Delimited Data File” plugin.

2.1.3 CSV File Collection

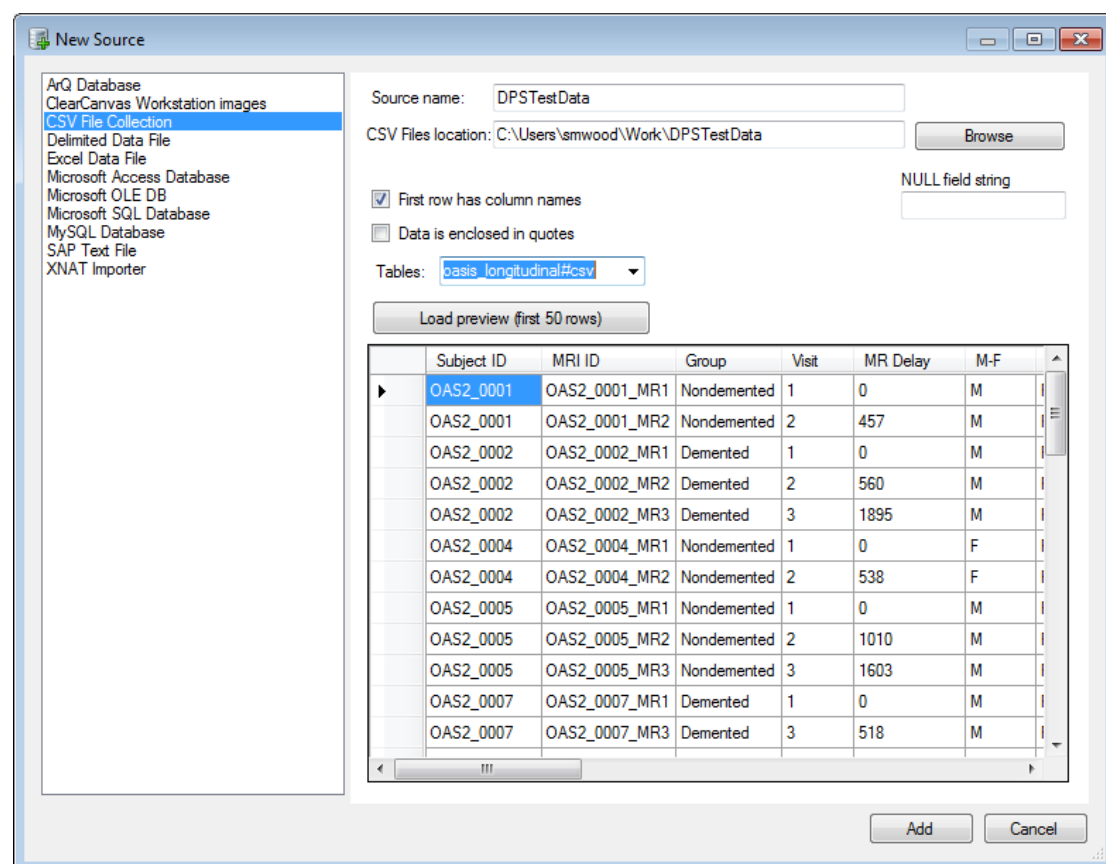


Figure 6 CSV File collection data source configuration

Much of this functionality has been discussed above. You browse to a folder which contains a number of CSV files and the system will load them all as separate tables. The viewing of the datable data and data typing issues are as above so will not be re-iterated here.



NOTE: The term CSV stands for Comma Separated File and this plugin requires exactly that. The term CSV is often abused and we commonly receive examples delimited file using semi-colons or tabs which have a CSV extension. If you have these types of data either use the Delimited Data File plugin if there is only one or transform the data yourself through something like Excel.

2.2 Database sources

2.2.1 Microsoft SQL, MySQL and ArQ

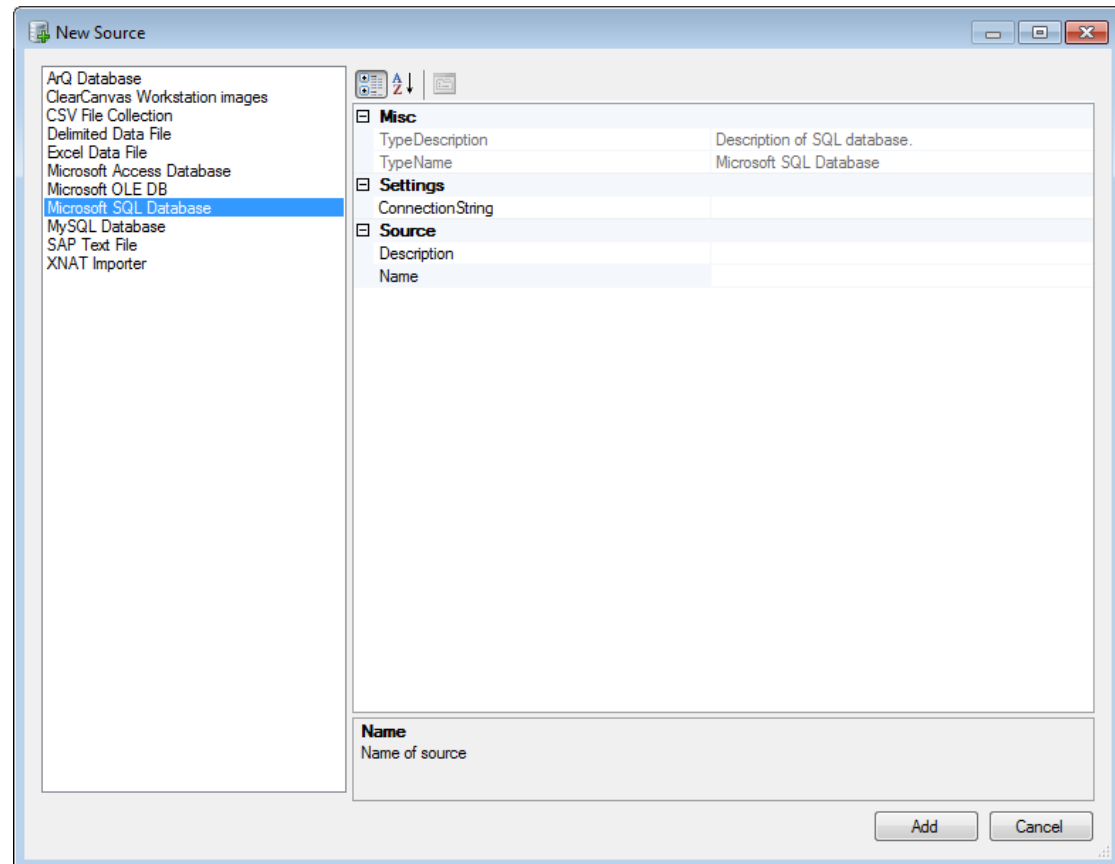


Figure 7 Database server data source configuration

These two plugins are identical from a users perspective and so will be discussed together. The configuration is rather unfriendly to the naïve user but would be obvious to any administrator with experience of managing these types of database. In essence there is only one option beyond the name and description of the data source and this is the “Connection String”. This is a well-defined term and we would normally refer users to the <http://www.connectionstrings.com> web site to find out which string to use for the specific database configuration they are using. An example of a connection string for a MySQL database running on the local machine would be:

Server=localhost;Database=crim1;Uid=root;Pwd=myPassword;

There are no data typing considerations with these types of plugin as they are picked up from the database themselves.

The ArQ plugin is just a specific version of the MS SQL database connector which has some bespoke processing embedded to maximise the utility of the data stored with an ArQ system.



NOTE: In most circumstances the connection string will contain the username and password to access the contents of the database and this has to be stored in the DPS project file for future use. The project files are not encrypted so you must ensure that this file is saved in a secure location where it cannot be

accessed by anyone who does not have legitimate access to the database contents.

2.2.2 Microsoft access database

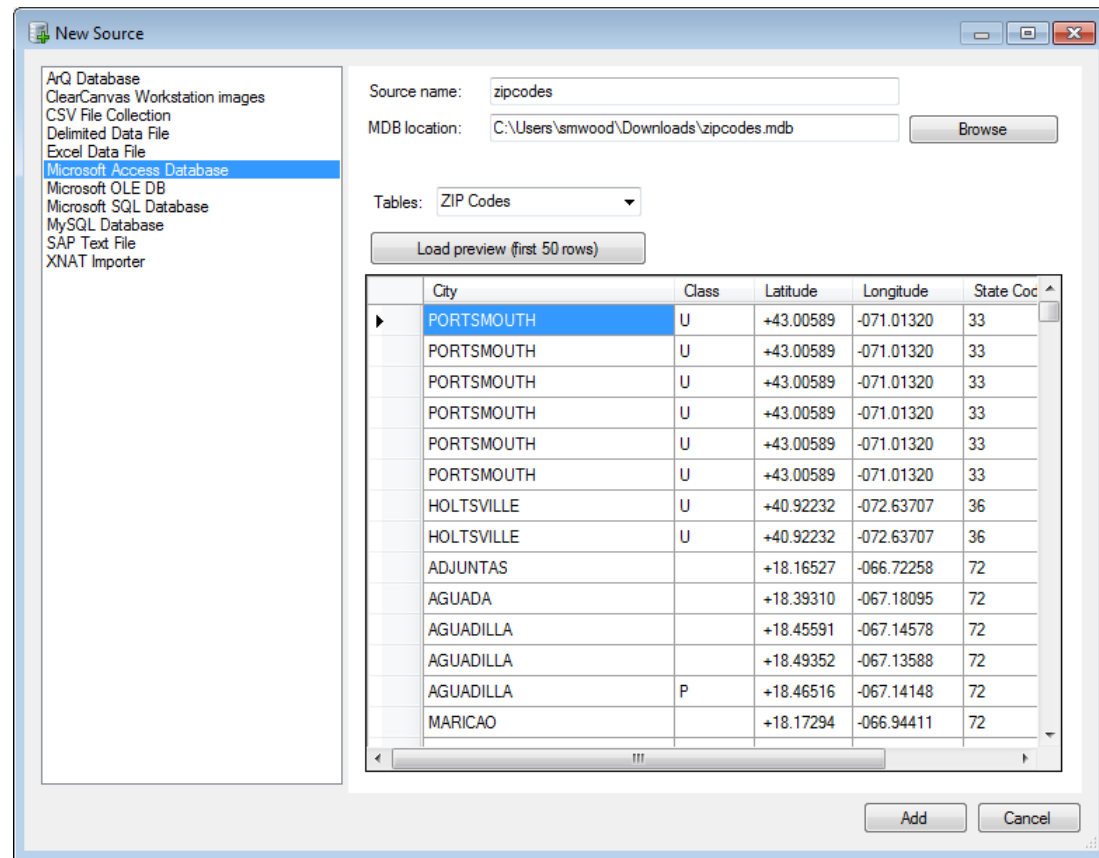


Figure 8 Microsoft access data source configuration

The user simply browses to the location of the MDB file and opens it. You can view the contents of the tables if you wish but there are no specific options associated with the tables and they will all be imported with the data source. As with the other database plugins data typing is not a problem nor should you need to define any relationships for the data source following import as these will be picked up directly from the database.



NOTE: Since this is explicitly related to Microsoft office it is entirely possible that some versions may not be supported by the plugin. Whilst we will endeavour to keep the system in step with developments from Microsoft it is possible newer, or indeed very old, MS Access databases may not load properly. As with the Excel plugin the fall back position would be to export the tables manually to CSV files and load them with the “CSV File Collection” plugin.

2.2.3 Microsoft OLE DB plugin

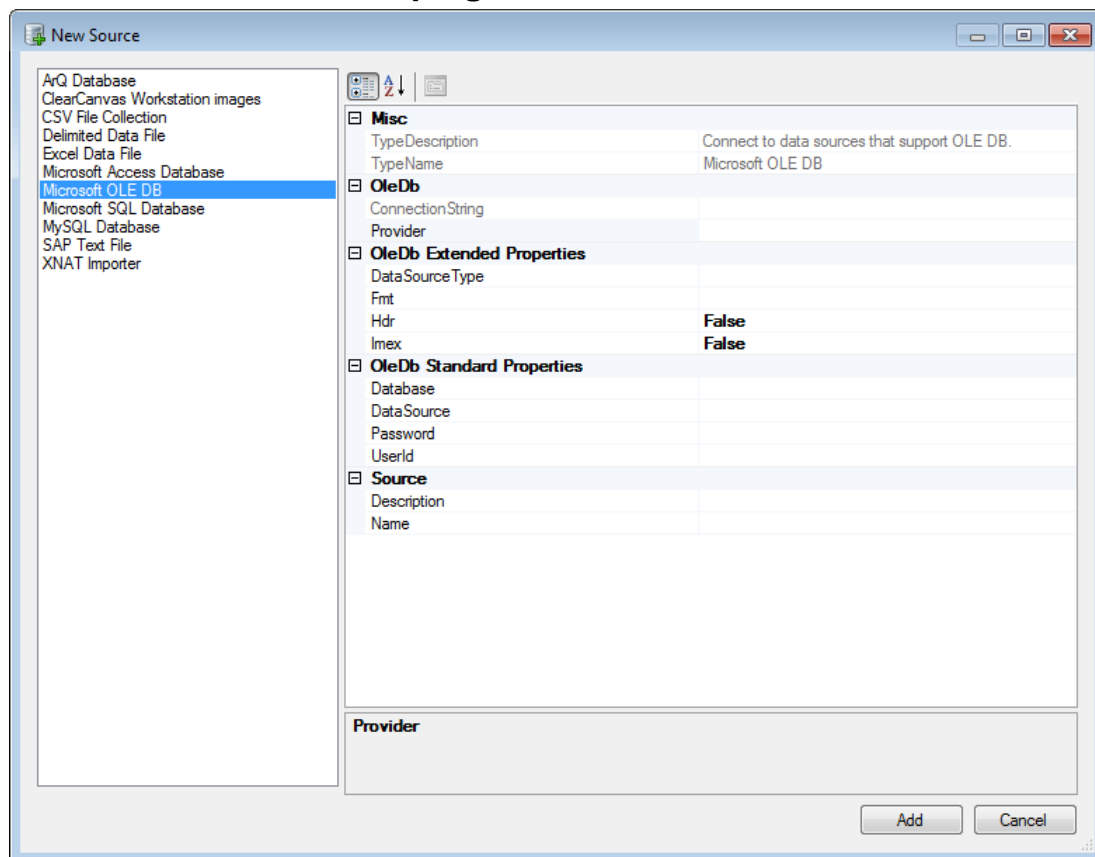


Figure 9 OLEDB data source configuration

This is perhaps this most unfriendly plugin for the standard user but in fact the most powerful from a data management perspective. It is capable of connecting to a very large range of data sources, and its capability is dependent on other components already installed on the operating system. For this reason it is actually not possible to give an accurate description of how it might behave on the users machine. The <http://www.connectionstrings.com/net-framework-data-provider-for-ole-db/> web site gives a good overview of the types of connections one might make with this plugin but there are many other internet based resource available if you search for “oledb”.



NOTE: We would not envisage that many users would attempt to use this plugin but if you are not able to get the data you have loaded into the DPS you can request support from the team and we will work with you to find a set of properties for this plugin that meet your needs in lieu of us developing a more bespoke and friendly plugin for that specific data source.

2.3 System specific sources

2.3.1 ClearCanvas Workstation images

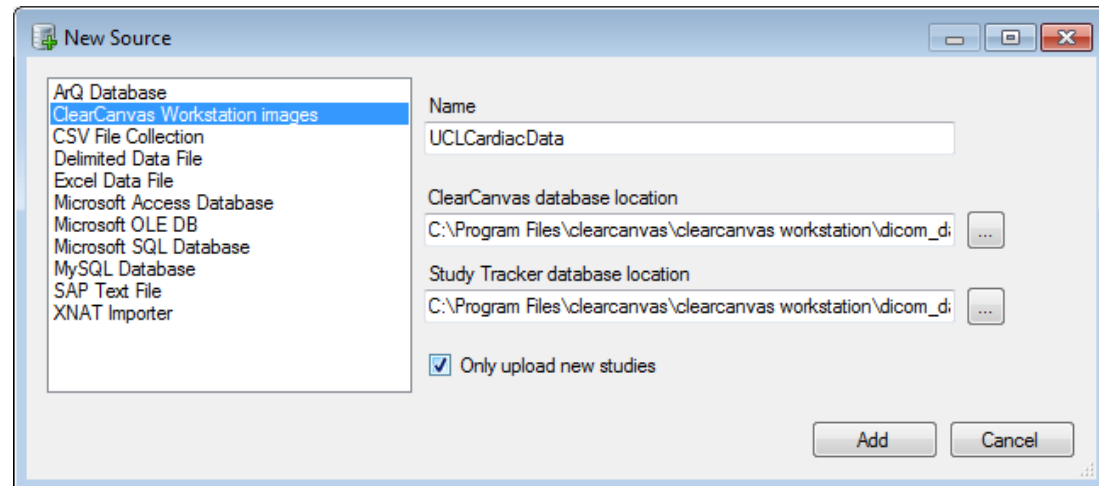


Figure 10 ClearCanvas DICOM data source configuration

This plugin provides a mechanism for managing DICOM images and depends entirely on a product called ClearCanvas Workstation. The installation files for a free version of this can be found on the VPH-Share portal.

In essence this can be used in two scenarios to benefit researchers in the imaging community. First it can be truly integrated with a PACS (Picture Archiving and Communication System) which would allow the institution to send specific studies (usually related to a research cohort) and then these can be processed and de-identified for publication to VPH-Share. This offers an opportunity to minimise the effort for clinical institutions in provisioning imaging collections to their research teams.

Second it is capable of ingesting and properly structuring large collections of DICOM files. In many institutions research data is stored on CD/DVD or a central filestore with little or no formal structuring or search capability. The ClearCanvas workstation can be used to import these collections and allow the users to achieve this structuring as well as offering an image viewing and processing capability.

The ClearCanvas workstation documentation can be found at http://www.clearcanvas.ca/Portals/0/ClearCanvasFiles/Documentation/UsersGuide/Workstation/2_0_SP1/ and will describe to the users how to use this as a DICOM image management system.

The usage of this plugin is in the extraction of the subject information from the imagestore and its publication as a coupled database/filestore for the DICOM metadata and image files. When used in conjunction with the ClearCanvas DICOM Upload plugin (see section on handling file references) the DPS can also de-identify the DICOM files and database records so images that were originally identifiable can be made available for secondary use.

The ClearCanvas database location is located under the installation folder, and by default can be found at:

C:\Program Files\clearcanvas\clearcanvas workstation\dicom_datastore\viewer.sdf

The location of the tracking database can be anywhere you wish as it gets created on import of source.

2.3.2 XNAT Importer

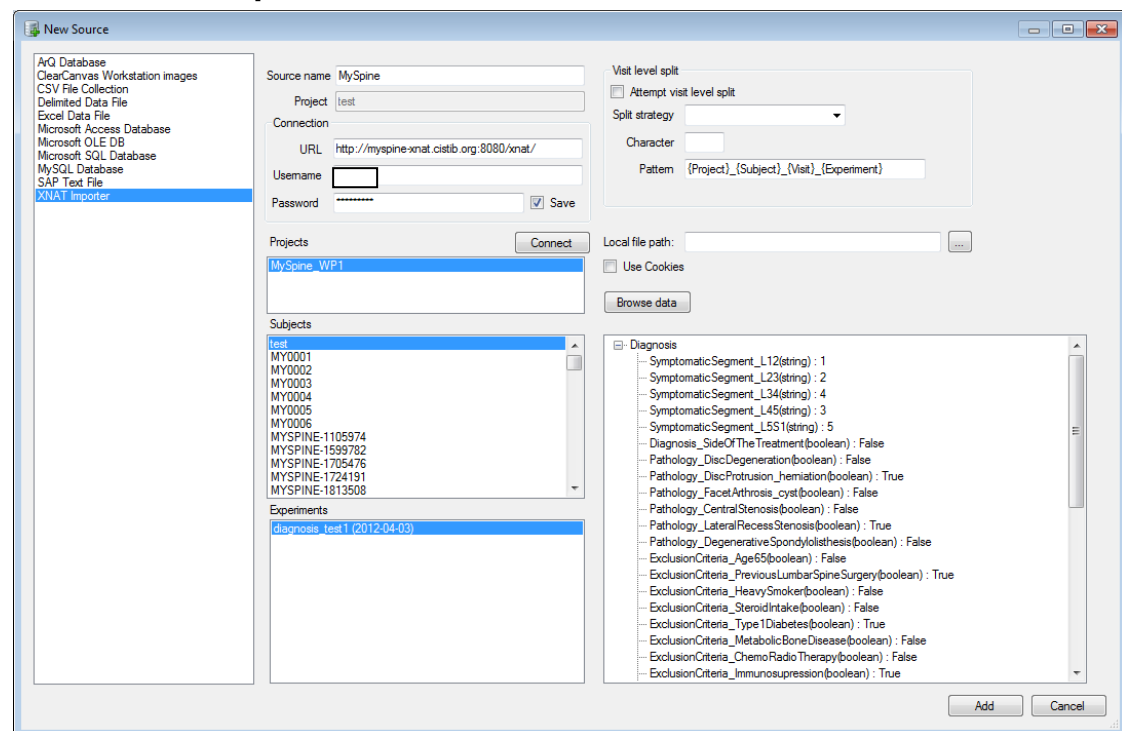


Figure 11 XNAT data source plugin

This plugin connects via web services to an XNAT instance and extracts data from it into a form that is easily queried. It also extracts all of the imaging data and uploads it to the cloud storage services adding in URL links to locations of the images as it does so. In many ways it utilises the PACS storage model only there is often significantly more subject level data.

In terms of what data can be extracted from the system we have gotten to the stage where the core data relating to the patients, visits and imaging studies can be extracted. XNAT does however have the facility to be extended to have custom data collection forms, but this process cannot be catered for automatically by the plugin. Because of this issue in every system we have connected to there has been some development work done on the XNAT Importer to effectively extract all of the data. By default the system will just extract that information it “knows” about so this will not prevent publication but if you require a more comprehensive integration please contact steven.wood@sth.nhs.uk to discuss how this might be achieved.

The importer has an advanced viewer which allows the user to browse the contents of the XNAT repository and get an understanding of what the exported data might look like. Often people who use the XNAT system are unaware of the complexity of the data model but when extracted into a relational database this becomes apparent and may be overwhelming.

The only real data the plugin needs at present is the URL of the XNAT server and some login credentials.

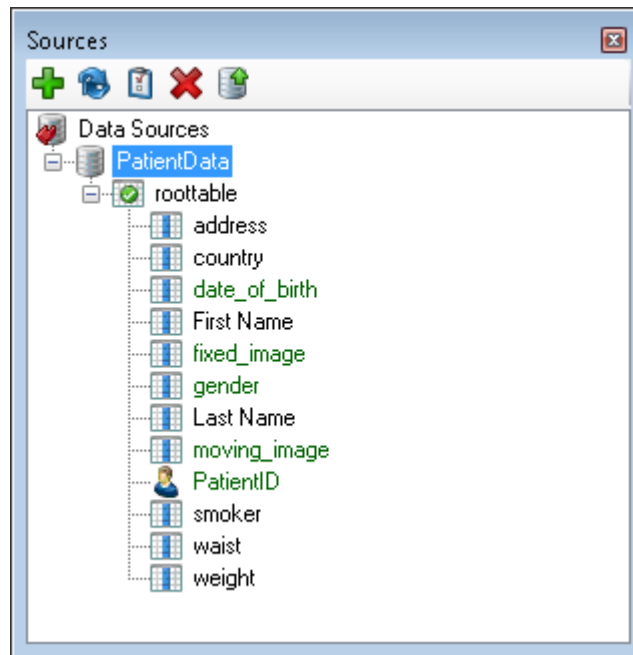


Figure 12. Source tree showing tables and fields.

3 Working with Tables and Fields

3.1 Source panel colour coding



The table has been linked into the rest of the tables, or there is only one table and it does not need linking.



The table is not linked and probably should be

Green text on any of the items in the data tree indicate that the field has a semantic annotation assigned

3.2 Renaming

Some dataset schemas will be easily readable by a human and can be used without modification; but some naming conventions are a little verbose (like the example data) and some have unrelated names such as “Field1”, so to make these easier to use the user can optionally rename the tables and fields without modifying the source. To modify a name the user should right click on the item which will display a context menu (see Figure 14) and click “Rename”. The name then becomes editable in the source tree view in a behaviour similar to the rename action in Windows Explorer.

After renaming the fields of our example data, it can be seen that there is a vast improvement in its readability. Figure 13 shows a diagram of all the tables of the example data and their renamed fields which are much easier to read.

NOTE: These modified names are used internally by the DPS, this process does not modify the source nor does it require the source to be manually modified externally.

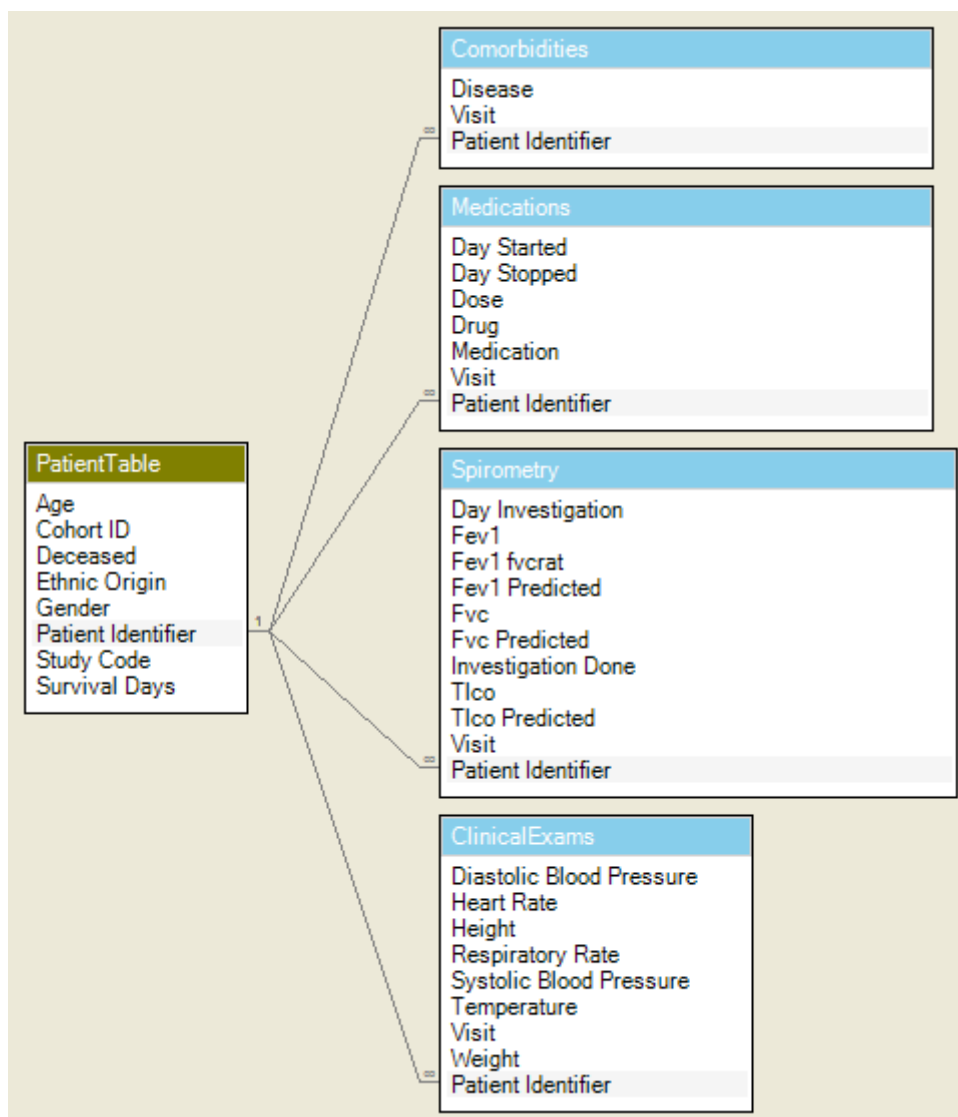


Figure 13. Relationships diagram.

3.3 Keys

A key is an item which can uniquely identify records. Within the DPS a key field within a table identifies a single row of data inside that table and a key table within a source uniquely identifies the top level record in the dataset.

If the source contains this information then it will be extracted as part of the schema but if the information is missing then you will need to set the keys manually. Since our example data is in a format which can't store keys, we will need to set them.

All the tables apart from PatientTable do not have a unique field to identify each row so key fields should not be set in these tables. During the publication process, any tables which do not have a key field will have one created automatically.

We should always be able to set the key table because for each dataset there will always be at least one table. The key table is the one which contains the rows of data which can be linked to all other data through as number of relationships (discussed in

the next section). Even though it seems like the key table could be automatically determined after the relationships have been defined, this is not the case as multiple tables could emerge as a key table, so it is left to the user to decide.

Setting an item as a key is simple, you should right click on the item (either a table or a field) where you will be presented with the context menu shown in Figure 14 and click “Set as key”. Since only one key can be allocated to a table or source if one has already been set it will be removed and the selected item set as the key.

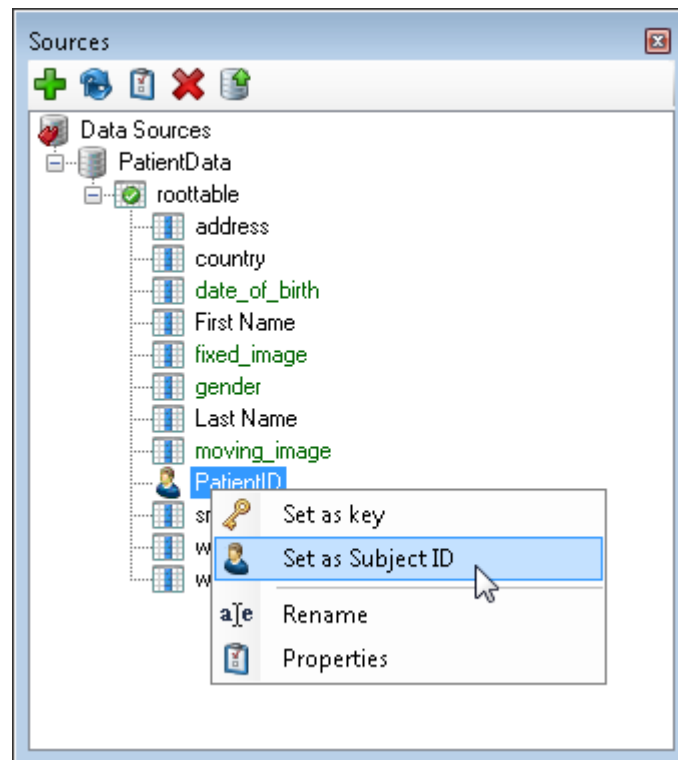


Figure 14. Source table right-click context menu.

3.4 Relationships

Relationships define how data is related between tables. These are very important as when they are missing data becomes orphaned and meaningless. A relationship informs the user or a query engine that a row in one table is linked to a row in another table when the values in the two fields match.

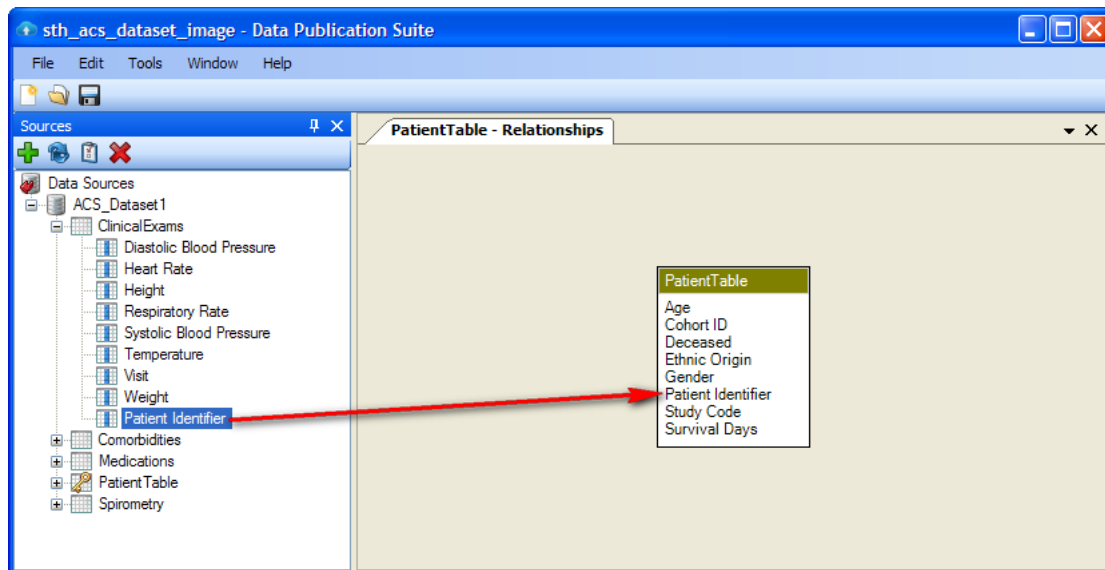


Figure 15. Showing the creation of a relationship by dragging one field onto another.

In the example data, each of the tables link to the “PatientTable” table via the “Patient Identifier” field. Figure 15 shows the creation of the relationship between the “ClinicalExams” and “PatientTable” tables.

To display the relationships window for a table, the user should right click on a table (see Figure 14) and select Relationships. When the Relationships window is displayed, fields from the Sources tree view can be dragged onto fields in the relationships window to create a relationship. Once a field is dropped into the Relationships window, a confirmation window is shown which lets you review the fields you have chosen for the relationship and confirm the type and direction of the relationship.

Figure 16 shows this conformation window displaying a preview of the relationship to be created. There are two types of relationship which can be created: one-to-one or one-to-many. The latter is the most common and is the type used for all the relationships in the example data. It means that each row of data in the left table (in the confirmation window) will link to zero, one or multiple rows in the right table. The direction of the relationship is indicted by the 1 and ∞ symbols on the line joining the tables, ensuring this direction is correct is important, if the tables are the wrong way round then you can click the “Swap Fields” button to switch their places. The direction of the one-to-one relationship is not important because it means that for each row in either table there can only be zero or one rows in the other.

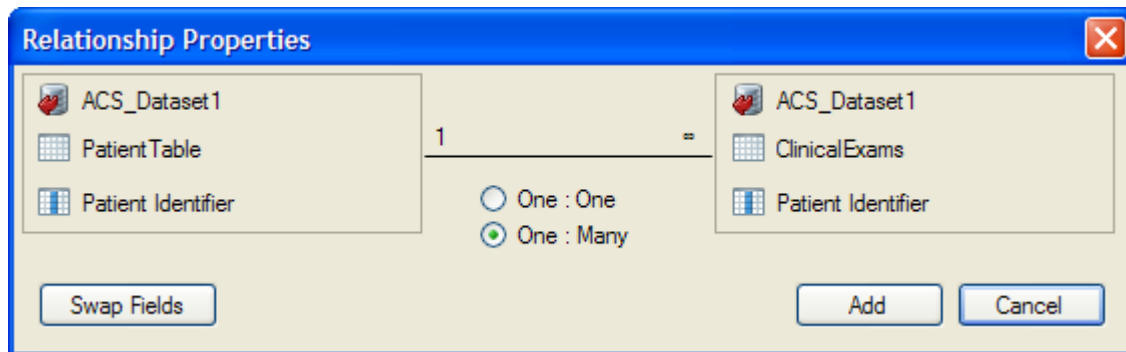


Figure 16. Relationship properties window is shown to confirm the direction and type of the relationship.

Once all the relationships have been created the relationships window will look like Figure 13. A relationship can be modified by double clicking its line between the tables in the relationships window where you will be presented with the relationship properties window; this is similar to the relationship confirmation window, the only difference is the addition of a delete button which allows the relationship to be removed.

4 Annotation

Annotation in general is the process of attaching some item of metadata, e.g. description, to a piece of data with the intention of turning into information. In this application the fundamental source for annotation is to use ontologies and this process is known as semantic annotation.

4.1 Semantic data annotation

Data annotation is the top level of annotation and simply assigns a concept to a data table or column within it. In terms of system functionality this step is optional in that you do not need to annotate a single element in the source data set. By default a new set of semantic terms based on the column names of the tables will be created and the data will be published with these into the SPARQL access point. This approach will make it very difficult for other users to search for, and find, the data set and will make the resulting data set almost impossible to interpret without further documentation but it is valid.

Assuming you wish to go further and annotate the data as fully as possible you can add annotations at the both the table and column level. This is done by using the ontology search panel. Simply type in the name of a concept you think represents the data and click search. You will then have returned a list of concepts that contain the keywords you entered.

NOTE: *Many concepts will have multiple terms from a number of different ontologies. Which one to select is still a subject for research and ultimately the system will rank the results in an intelligent way but this does not exist yet.*

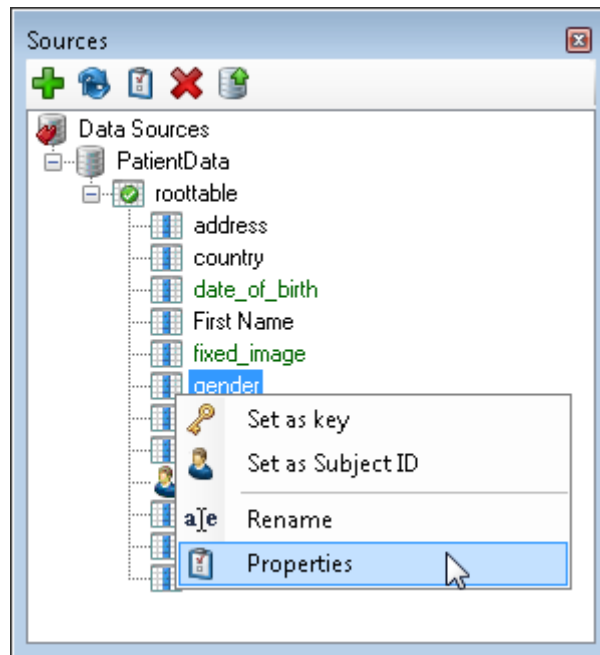


Figure 17 Display annotation properties for a source field

Once you have searched for a term adding it to the table or data field is as simple as dragging it onto the item in the source tree, or dropping it into the box on the field properties window as shown in Figure 18.

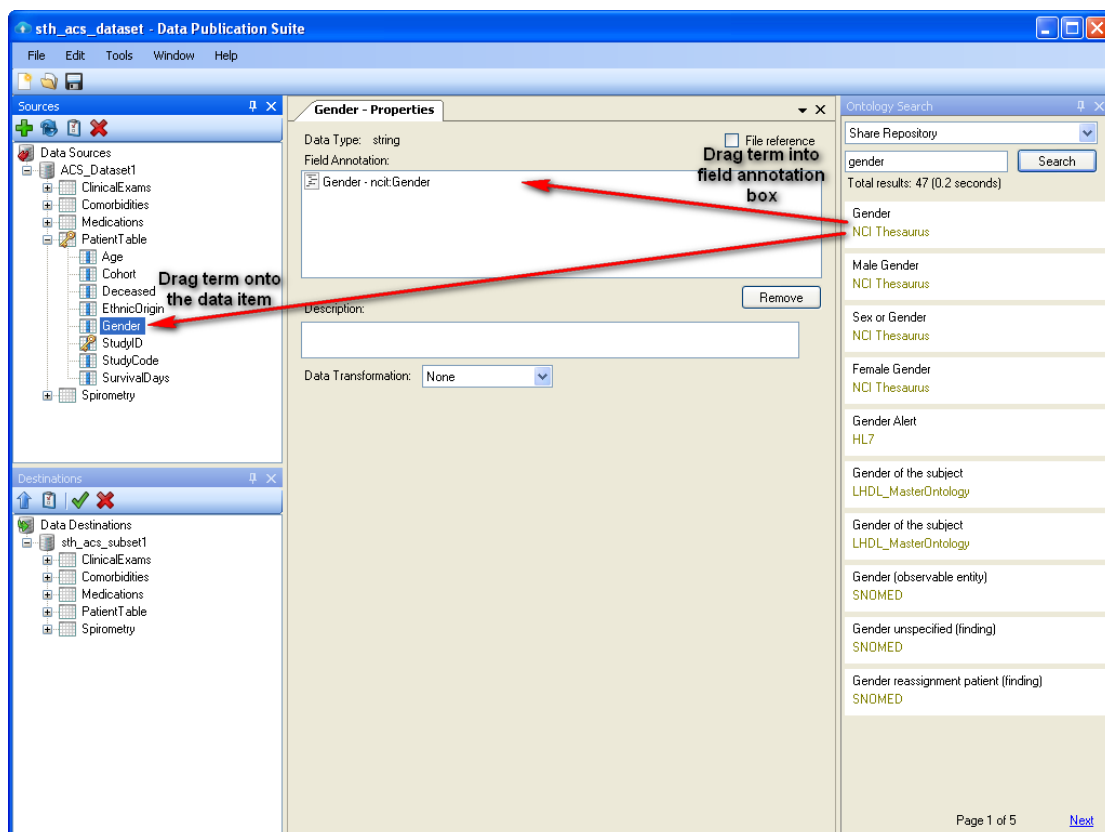


Figure 18 Annotating data items with ontological terms

Should it not be possible to find an ontological term that accurately describes the data item you have the current fall back is to add a free text description of the data so this can be passed through to the person consuming the data.

4.2 Semantic data transformation

Sometimes, simply annotating the column of data is not enough to fully define the meaning of the actual data held in it. For instance we have annotated the data field in 4.1 with the concept Gender. This however is not helpful if the contents of the field are simply 0 or 1 with no indication of which one relates to male and which to female. To help with this issue we have created facilities for transforming the actual data within the fields to concepts as well in order to help with general understanding and querying the data.

The simplest form of this process is to create a new transformation collection as shown in Figure 19. Once we do this the software analyses the data in the field and returns the unique values so we can now add a more formal definition of what each element means.

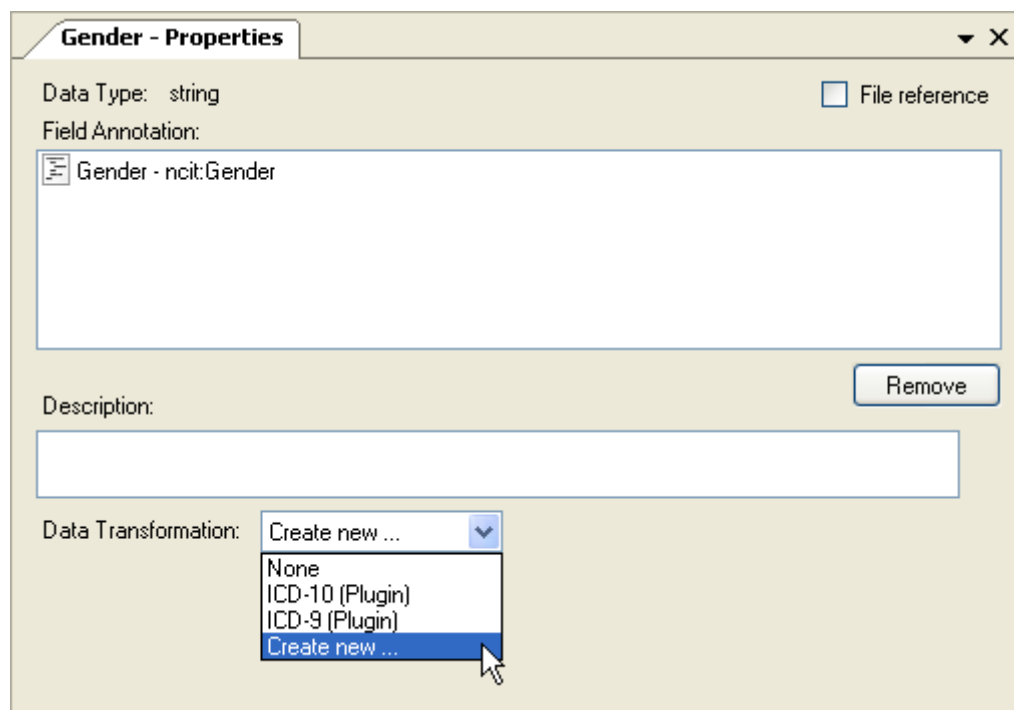


Figure 19 Creating a new data transformation collection

Gender - Properties

Data Type: string ☐ File reference

Field Annotation:

Gender - ncit:Gender

Description: Remove

Data Transformation: Gender

Values:

1
0
--EMPTY--

☐ None ☐ XSD Type ☒ Ontological

Male
NCI Thesaurus

Figure 20 Annotating a single data instance

If you then select one of the values you can choose to transform it into a standard data type such as a number or date, or define it as an ontological concept. Figure 20 shows the latter option and as usual you simply drag the selected concept into the box from the ontology search results.

Figure 19 also shows the fact that you can write your own plug-ins to perform this data transformation. An example being that a dataset with a column of diagnosis codes could contain hundreds if not thousands of unique values. Annotating these by hand in the way just described is not practical so the DPS offers a way for people to develop the functionality in this area and add some automation to it.

With the completion of the annotation process the data source is now completely defined and we are ready to decide how to publish it for access by other users.

4.3 View the table contents

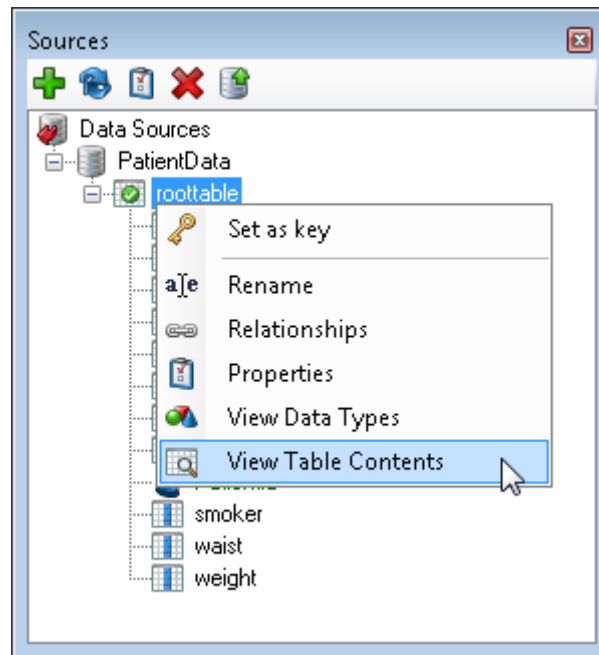


Figure 21 View table contents option

This will simply display the table contents unaltered from the source.

4.4 View Data types

When dealing with text based file sources it is especially important to view what the system thinks the file types are. As described earlier there are many ways in which this interpretation can be performed and inspecting this list may prompt you to re-export the data in a slightly different format if problems arise.

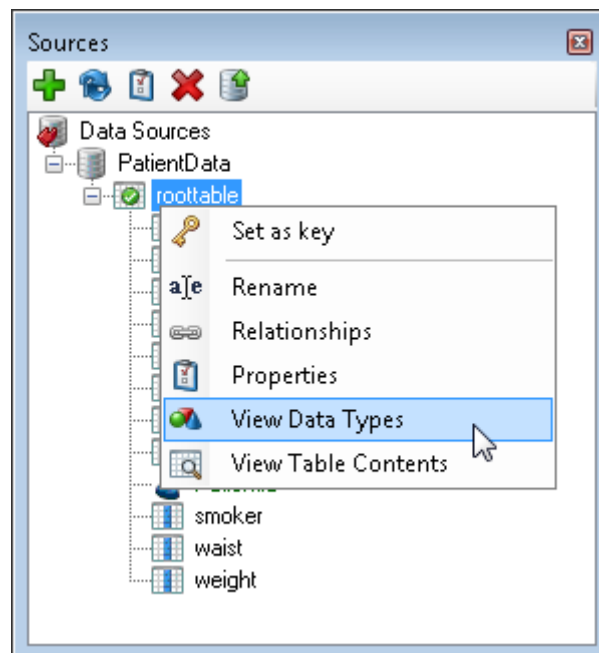
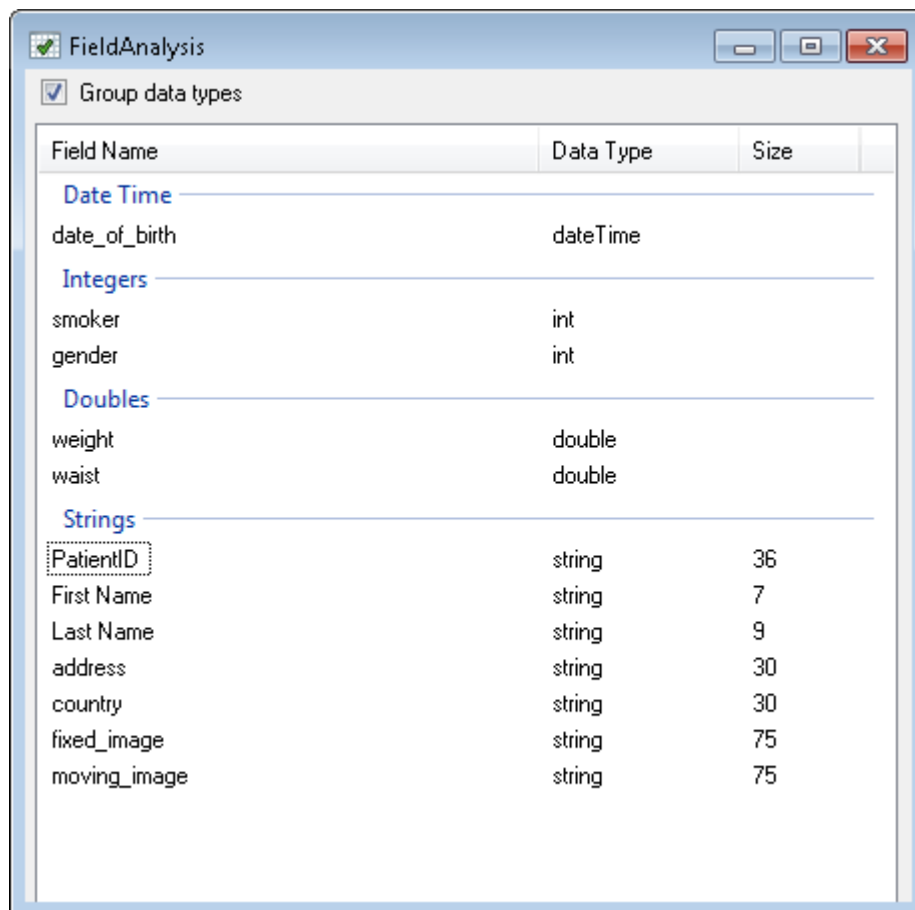


Figure 22 Display data types option



The screenshot shows a window titled "FieldAnalysis" with a checkbox "Group data types" checked. Below this is a table with three columns: "Field Name", "Data Type", and "Size". The table is organized into sections: "Date Time", "Integers", "Doubles", and "Strings".

Field Name	Data Type	Size
Date Time		
date_of_birth	dateTime	
Integers		
smoker	int	
gender	int	
Doubles		
weight	double	
waist	double	
Strings		
PatientID	string	36
First Name	string	7
Last Name	string	9
address	string	30
country	string	30
fixed_image	string	75
moving_image	string	75

Figure 23 Display of data types assigned to each column of the table

5 Creating a new destination

At this point standard users can not create a new data container on the VPH-Share data nodes, this has to be done by the system administrator (at this point please mail steven.wood@sth.nhs.uk for support). Once you have requested, and had create, your own data instance the process for populating it with data is as follows.

First, right click on the Source you wish to publish and select “Add as new destination”

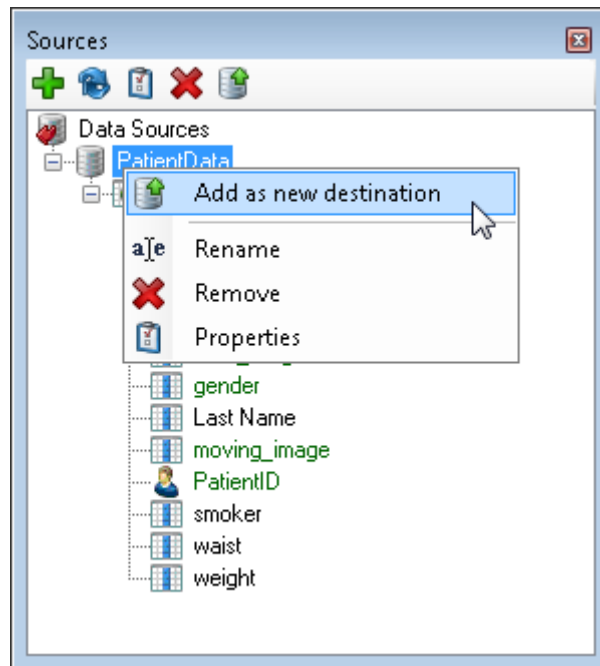


Figure 24 Adding a new destination from a source

If you have not already logged into the system you will be asked at this point, after which you will see a list of all data instances on the server. Select the one created for you and press OK.

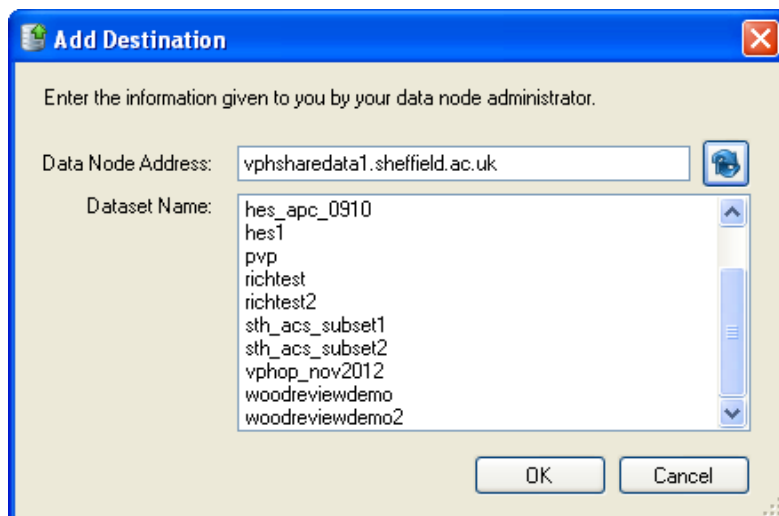


Figure 25 Connecting to a datanode and selecting a publication container

Within the destinations section of the interface you will see a mirror image of the table and field structures shown in the sources section but the options applied to each of these items are now quite different. This is the area where the de-identification profile is created for a given instance, note there can be many destination for the same source which is the equivalent of creating multiple views of the same data set to different users or groups.

6 De-identifying the dataset

There are many options for de-identifying the dataset, in particular you can write your own tools and embed them into the application for handling specific data types so it is not possible to give examples of the functionality you may find in any given installation. We will however go over the core components and some examples of the freely provided tools to give an idea of the process.

6.1 Table options

There are only 2 options for tables, these are either Include (the default) or Exclude.

NOTE: *If you exclude a table it will override any options applied the fields within it and no data from the table will be published.*

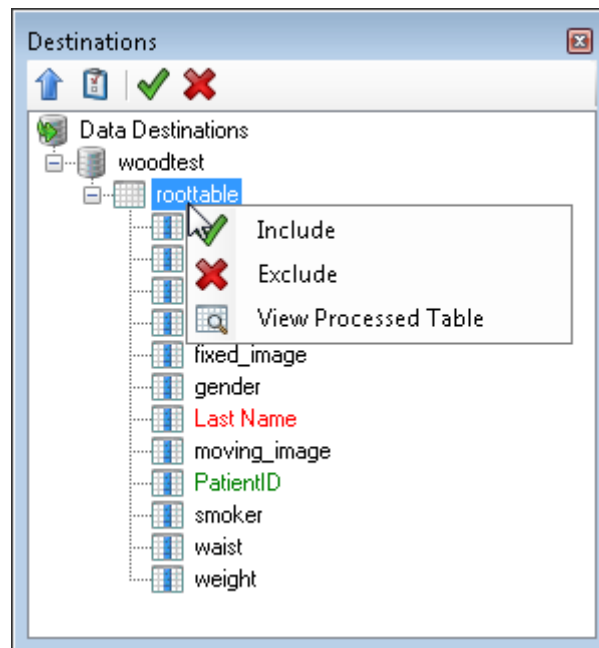


Figure 26 De-identifying table options

6.2 Field options

As with tables the options to simply publish unaltered (Include) or withhold from the publication (Exclude) exist on every field. However there is an addition option with is the Properties item.

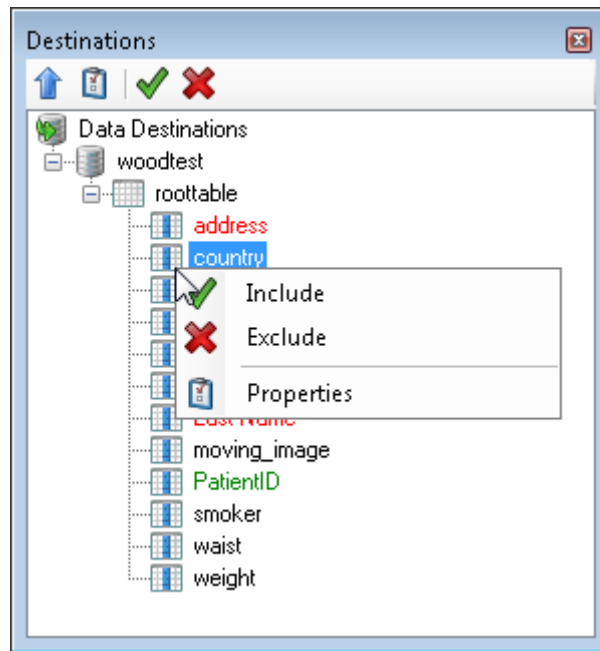


Figure 27 De-identifying data field options

When selected you will be presented with a properties window in the middle area of the interface that allows you to transform this data item.

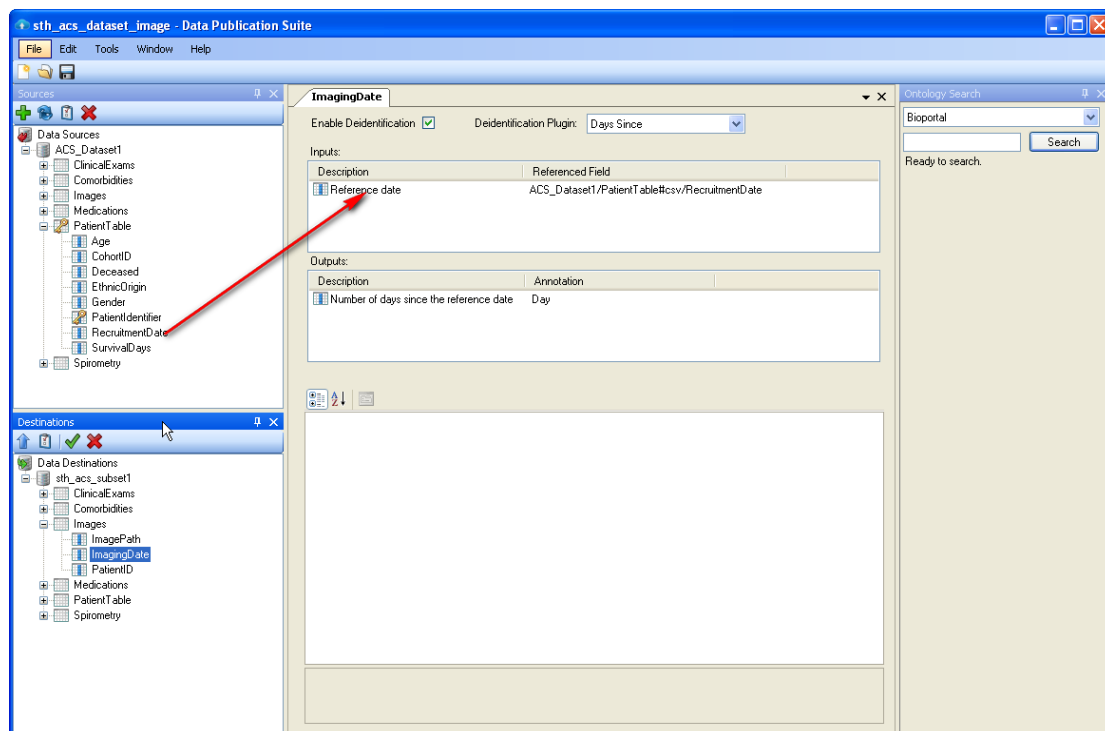


Figure 28 Advanced data field processing options

The reason this looks complex is that there are many ways in which one might de-identify a data item. Indeed often the resulting piece of data is not the same as the one annotated in the source. For instance you may wish to change date of birth to age but now we would explicitly want the output data item to be annotated with the new concept to ensure it meaning is correct. The process may be even more sophisticated than that, the example above shows the use of the Days Since plug-in. Here we not

only have a different output type but the plug-in also requires some other information, in this case a date on which to calculate the elapsed days. Above we have chosen the recruitment date as the datum for the calculation which is a very common process in clinical trials for de-identifying the dates of events.

6.3 File options

To be presented with the file properties window you would have selected the checkbox on the source data item indicating that this field contains a file or folder path on the local, or potentially networked, drive. The options on this page describe how this is to be handled.

The first decision is if the file references should be uploaded at all. Assuming you check the enable file upload box the next two controls become enabled. First you must select a folder in the LOBCDER filestore to upload the files to.

There is an option for using a plug-in to process the file or folder before uploading it to the network. Since these can be produced by anyone it is not possible show a list of available options or what the interface may look like once one is selected but we will use the example of the DICOM file processor supplied with the DPS to outline the process.

The screenshot shows a dialog box titled "DestinationFileFieldProperties". It contains several sections for configuring file upload options:

- Enable file upload:** A checked checkbox.
- Use file processing plugin:** A checked checkbox with a dropdown menu set to "DICOM".
- Root upload folder:** A text field containing "/STH/ReviewDemo" and a browse button ("...").
- Metadata:** A section with the following fields:
 - Name:** A text input field.
 - Category:** A dropdown menu.
 - License:** A dropdown menu.
 - Description:** A large text area with scrollbars.
 - Status:** A dropdown menu.
 - Tags:** A text input field with an "Add" link below it.
 - Semantic Annotations:** A text input field.
- Plugin Properties:** A section with three rows, each containing a label and a text input field with a pencil icon:
 - PatientId
 - PatientName
 - TemporaryDirectory

Figure 29 File processing properties window

At the bottom of the window is an area titled Plugin Properties which is generated by the plugin itself. In this case it is asking for a Patient ID, a Patient Name and the location of somewhere on the local disk where it can temporarily store the processed files before they are uploaded to the LOBCDER services.

The other two fields can either have simple text put into them, in which case this information will be used in every instance of the process over all records in the dataset. However, you can drag & drop any data item from the data sources area into these fields. Once this is done this data item will be used during the processing of every file associated with the current record, and if the field in question has also been de-identified or encrypted the new value will be used in the process.

The section on metadata is a place holder for now and is not functional in this version of the DPS. In time this will be the metadata that is attached to the files or folders once they are uploaded to the network.

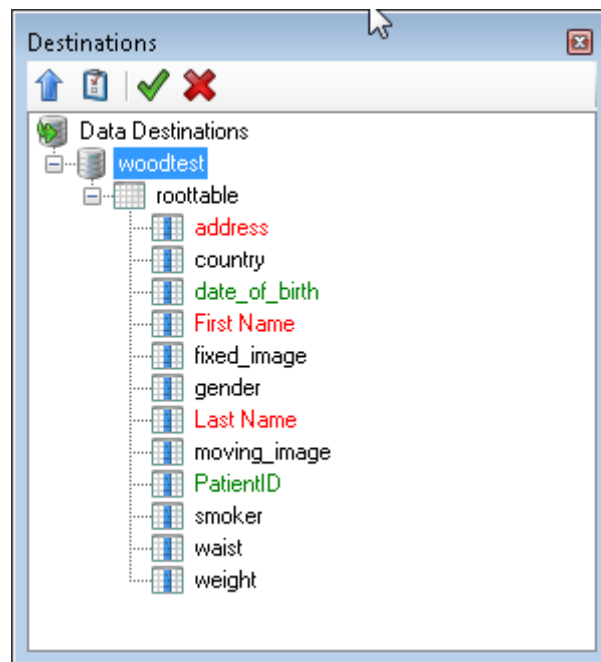


Figure 30 Colour coded displays on field items

As shown above, fields are colour coded to quickly indicate which of the 3 options have been applied. Red= Excluded. Black=Include and Green=Processed.

7 Dataset Properties

Figure 31 shows how to get to the destination dataset properties.

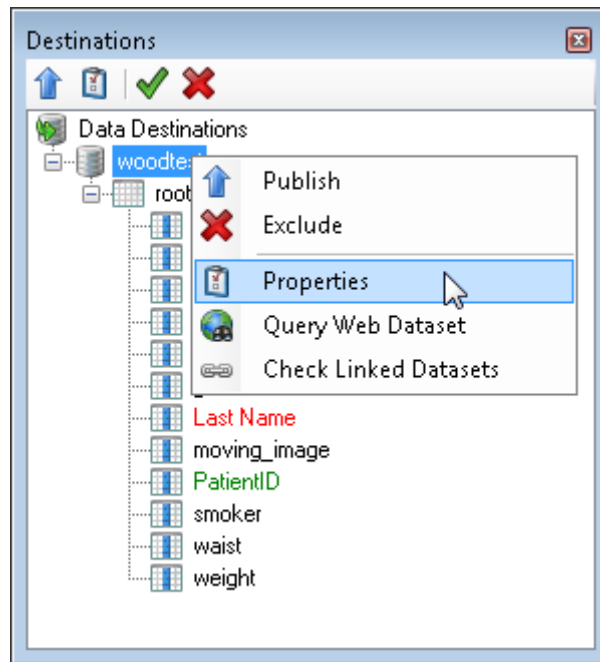


Figure 31 Destination data set properties menu

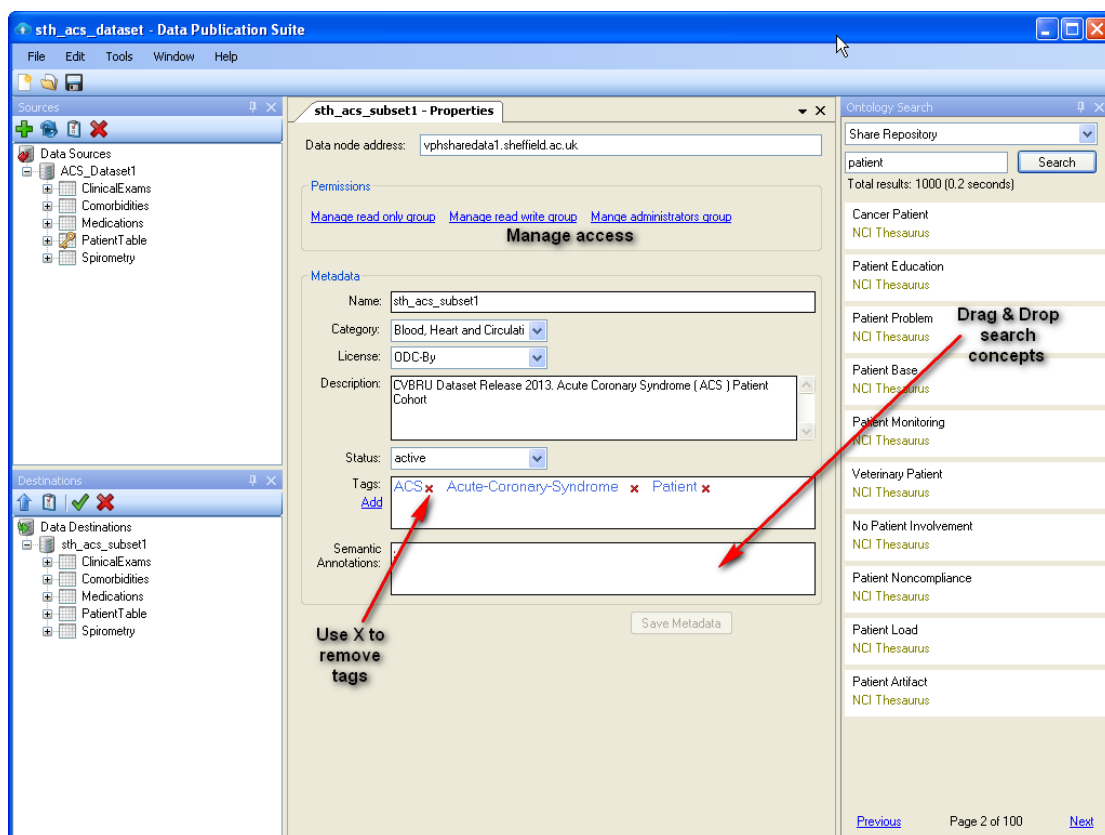


Figure 32 Dataset properties

The properties panel for a destination dataset contains two main components the area used to display and update the global metadata for the resource and the access control for the resource.

7.1 Metadata/Search properties

The primary use of the metadata added onto any resource is to enable data discovery. With this in mind there are three primary fields that support this, the free text description, the list of tags which is essentially a list of keywords and the semantic annotations. The name of the resource can not be changed at this point but is displayed for convenience.

The metadata on any resource can also be modified from the master interface.

7.2 Access control

Associated with every published dataset are 3 roles:

- Data Read Only
- Data Read Write
- Data Administrator

The management process is the same for each of the roles so we will pick the Read Only role and show how it can be managed from within the DPS. It should also be noted that the management these roles can be performed from within the master interface.

Figure 33 shows the form that allows you to add or remove individual users or groups from the read only role on the selected dataset. The form contains two areas, the current access list at the top and the search results for user or groups that can be added at the bottom. You can enter any text into the search box, which will be used as part of a “contains” query on users, groups or both as dictated by the options in the menu item.

Once you have found the item you wish to add highlight it and click the Add Item hyperlink and it will be applied. To remove a user or group simply click on the remove hyperlink next to the name in the top window.

There is no save on this form, any changes made take effect immediately.

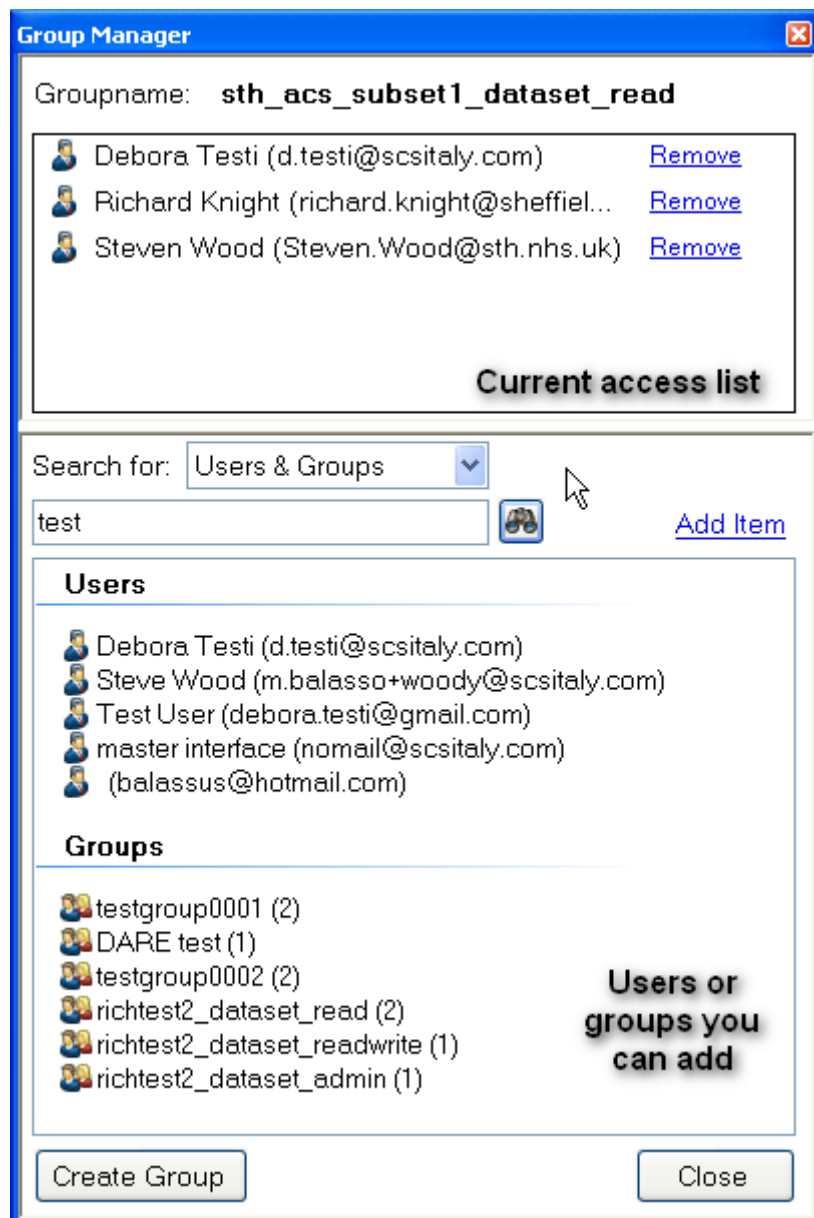


Figure 33 Managing access for a role on the dataset

8 Data publication

At this point all that's left to do is publish the data set to the server. This is shown in the context menu in Figure 34.

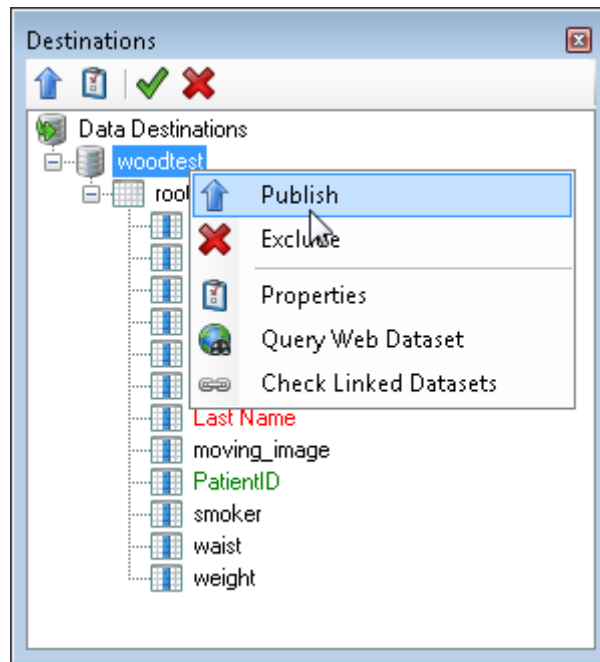


Figure 34 Publish the destination dataset

This will do everything necessary to upload the data to the server and whilst the process is in progress you will be shown a window indicating its progress. Please note that some of the steps can be time consuming especially Uploading Data which is heavily dependant on your network connection.

NOTE: Whilst the client side process can complete relatively quickly it can still take several minutes for the server to complete its work and upload the metadata etc across the VPH-Share network. This means that repeating the publish process in quick succession may fail but you should be informed that a process is already underway.

9 Query the Web Dataset

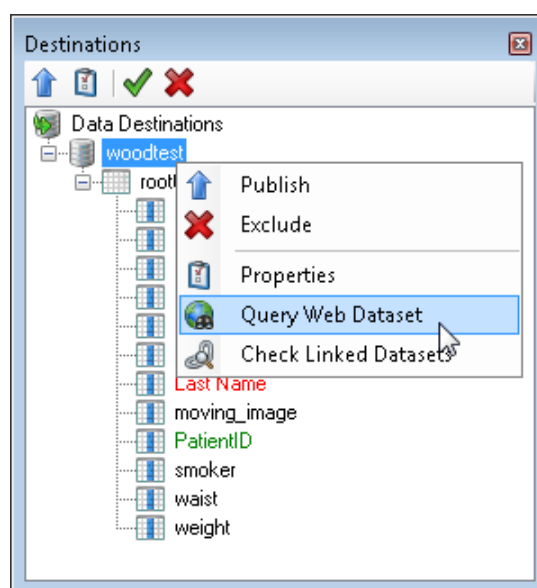


Figure 35 Query the published dataset

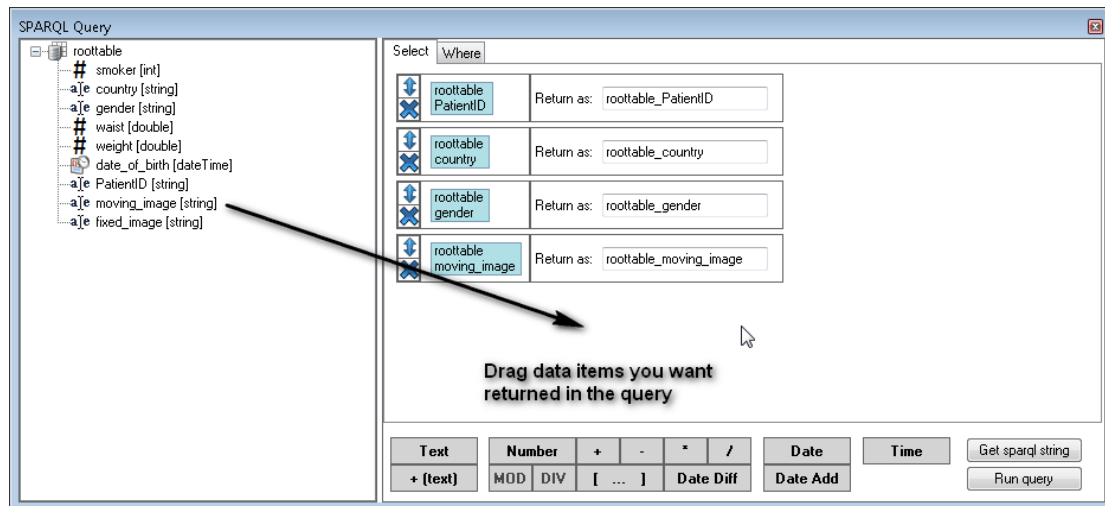


Figure 36 Start to build a SPARQL query

When we run this query with the “Run query” button we get the following:

The image shows the Results window with 10 records. The table has the following columns: roottable_PatientID, roottable_country, roottable_gender, and roottable_moving_image. The first record is highlighted.

roottable_PatientID	roottable_country	roottable_gender	roottable_moving_image
RRHMIU2qV2Uco9C/E9/nUhgA=	Virgin Islands, British	Male	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0390.dcm.dcm
8uk6ZFLxGQ1BxCezVgxuuqLE=	Samoa	Male	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0400.dcm.dcm
nIPF979pRagHMUGMnSiiakKNMeA=	Slovenia	Female	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0427.dcm.dcm
KUyww/F60KyyDx20reEZiPHZTIM=	Benin	Female	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0429.dcm.dcm
MGpFSDGTzCrCb3Xp3wA0c2fwWD0=	Guinea-Bissau	Male	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0431.dcm.dcm
o5sBdZVEPoLAWXjckCFqGZqBkA0=	Turkmenistan	Male	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0325.dcm.dcm
/PLXF2HHJwLFx3PkstH0Y0UUV8=	Jordan	Female	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0408.dcm.dcm
/1Caddh53zzn1k8PAjYjlduvw=	Taiwan	Female	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0436.dcm.dcm
I3bVDw3HHgkNmqbXKP0a/Azbo9U=	United Kingdom (Great Britain)	Female	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0442.dcm.dcm
cFGpPreuSxIFjH0V3HNNFvEjdlL=	Nepal	Male	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/IM_0383.dcm.dcm

Figure 37 Simple query results from SPARQL Service

The above query has no constraints on it so is a list of all rows in the datasource. If we now click on the “Where” tab we can add data items again but this time we are offered the option of placing values and comparison types on them. Figure 38 shows that we have added the country field and selected CONTAINS the letter “Brit”. The results of this query are shown in

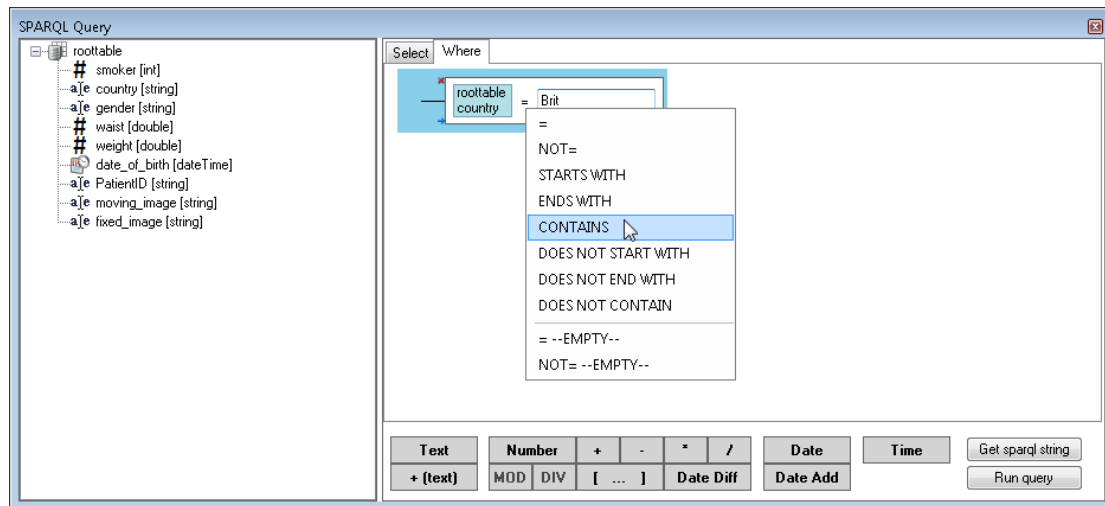


Figure 38 Adding constraints to the query

Results window

Number of Records: 2

	roottable_PatientID	roottable_country	roottable_gender	roottable_moving_image
▶	IRRhMtU2qtV2Uco9C/E9/nUhgA=	Virgin Islands, British	Male	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/
	I3bVDw3HHgkNmqbKKP0a/Azbo9U=	United Kingdom (Great Britain)	Female	https://lobcder.vph.cyfronet.pl/lobcder/dav/home/woody/demo/
*				

Figure 39 Results of query with constraints

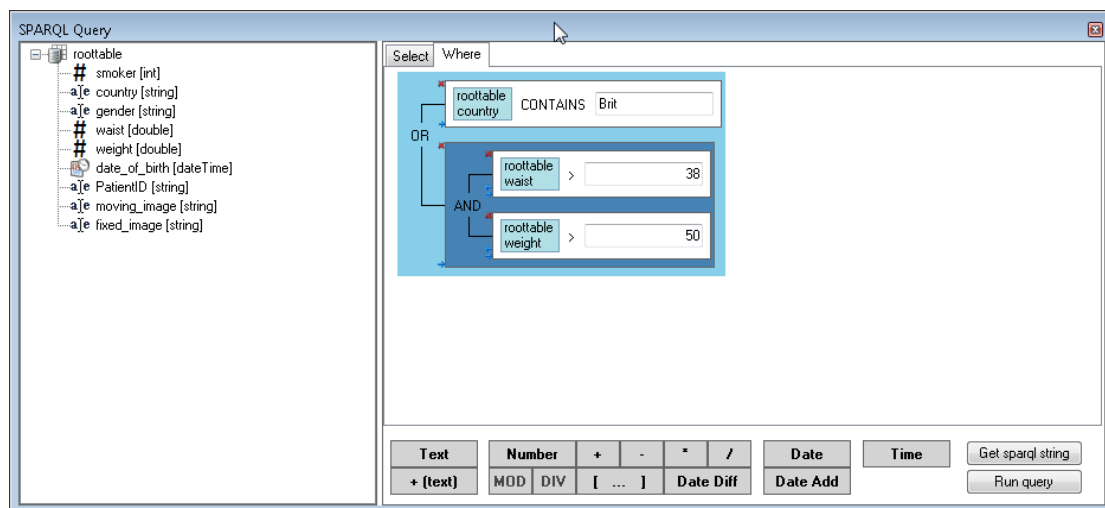


Figure 40 Complex query formulation with sub-groupings